Available online at http://www.ijasrd.org/in

International Journal of Advanced Scientific Research & Development

Vol. 01, Iss. 01, Ver. I, Oct – Dec' 2014, pp. 49 – 54

e-ISSN: 2395-6089
p-ISSN: 2394-8906

Effective Service Composition Using Semantics-Based Automated Service Discovery

S. CHITRA

Assistant Professor, V.R.S. College of Engineering and Technology, Villupuram, Tamilnadu, India.

A. KARUNAMURTHY

Ph.D., Research Scholar, Bharathiyar University, Coimbatore, Tamilnadu, India.

ARTICLE INFO

Article History:

Received: 14 Dec 2014;
Received in revised form:
21 Dec 2014;
Accepted: 24 Dec 2014;
Published online: 30 Dec 2014.

Key words:

Service composition,
Services discovery processes and methodology,
Web services discovery,
Web services publishing.

ABSTRACT

Composition of Web services has received much interest to support business-to-business or enterprise application integration. On the one side, the business world has developed a number of XML-based standards to formalize the specification of Web services. In this paper, we propose the issue of web service discovery given non-explicit semantic-based service request. Our approach of semantic-based web service discovery involves semantic-based service categorization and semantic enhancement of the composition of service requests. And the functional level service categorization is based on ontology framework. Also uses clustering, to classify the web services accurately based on the service functionality. The semantic-based categorization is done offline at UDDI. The semantic enhancement process includes better matching with relevant services. Additional terms related to service requests are retrieved from the ontology. Service matching between requested service and the retrieved service description is done through Latent Semantic Indexing (LSI) method. Our experimental results validate the effectiveness and feasibility of the proposed scenario.

INTRODUCTION

Web services are considered as self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Nowadays, an increasing amount of companies and organizations only implement their core business and outsource other application services over Internet. Thus, the ability to efficiently and effectively select and integrate inter-organizational and heterogeneous services on the Web at runtime is an

How to cite this article: Chitra, S., & Karunamurthy, A., (2014). "Effective Service Composition Using Semantics-Based Automated Service Discovery". *International Journal of Advanced Scientific Research & Development (IJASRD), 01* (01/I), pp. 49 – 54.

important step towards the development of the Web service applications. In particular, if no single Web service can satisfy the functionality required by the user, there should be a possibility to combine existing services together in order to fulfill the request. This trend has triggered a considerable number of research efforts on the composition of Web services both in academia and in industry.

A web service is a self-described application that uses standard Internet technologies to Interact with other web services. Most of the web service discovery approaches searches for web services that have semantic tagged descriptions through various approaches such as OWL-S, Web Service Description Language (WSDL)-S. It is impractical to expect that all new services having semantic tagged descriptions. Already existing web services are not having associated semantics.

The service description for WS1 is "This web service returns historical weather information for a given US postal code, date, and time." With inputs as Postal Code, Date, Time, and outputs as Temperature, Humidity, Pressure, Precipitation. WS2 is described as "Describes city information for a specific US city and state." With its input parameters City, State, and output parameters as Population, Temperature, Wind, Precipitation. In addition, the user formed service request may not include all the relevant keywords for discovering all the appropriate services within the UDDI. For example, the user may search for a web service stating "Find the temperature and rainfall based on zip code." However, there may be services published that provide relevant information based on regions, city names, addresses. These services could be combined with other locator services to yield better results.

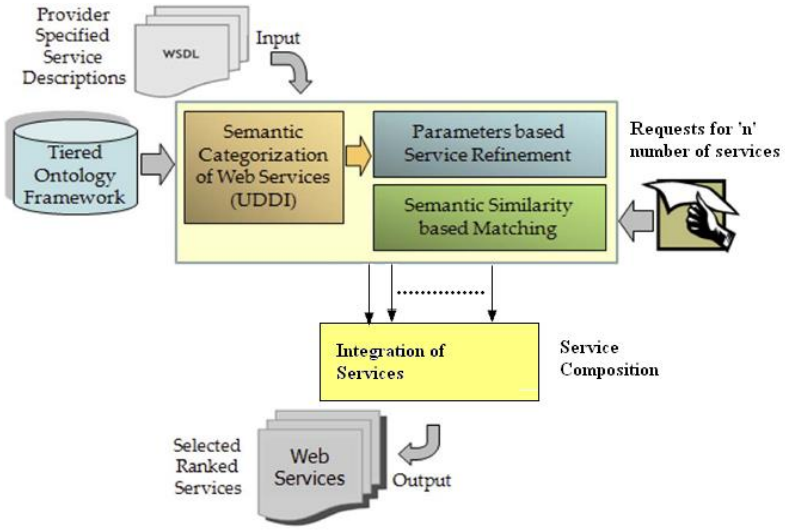

Fig. 1. Effective Service Composition using Semantics-based automated service discovery.

BACKGROUND

In the existing system the methodologies utilized for Semantics-based automated service discovery are,

(i) Semantic categorization of web services
(ii) Parameters-based service refinement
(iii) Semantic similarity-based matching

Semantic categorization of web services method includes the steps for categorization as follows:

- Web Service Vector Formation
- Web Service Vector Modification
- Clustering and Ontology Concept Association
- Web Service Registry—Reliance on UDDI

Parameters-based service refinement method includes the steps for refinement as follows:

- Service Parameters Retrieval
- Hyper-clique Pattern Discovery
- Ranking Semantic Associations
- Association Pattern Collection Pruning

Semantic similarity-based matching method includes the steps for matching as follows:

- Service Request Preprocessing
- Service Description Retrieval
- Ontology Concept Acquisition
- Service Request Expansion and Term-Document Matrix Formation
- SVD Transformation
- Service Request Projection

PROPOSED APPROACH

In this paper, we propose the additional concept of composition of more than one services requested by the clients at the same time. Finding out of the new searched services based on semantic-based matching are carried out by the methods of existing system. But the composition of found services is done by integrating them by using the one of the composition flow models.

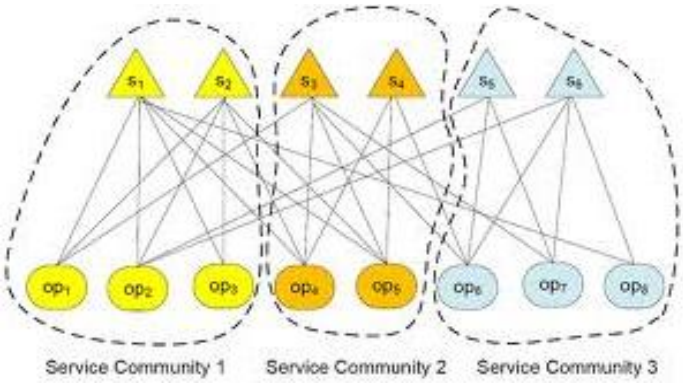

Fig. 2: Methods of service combining

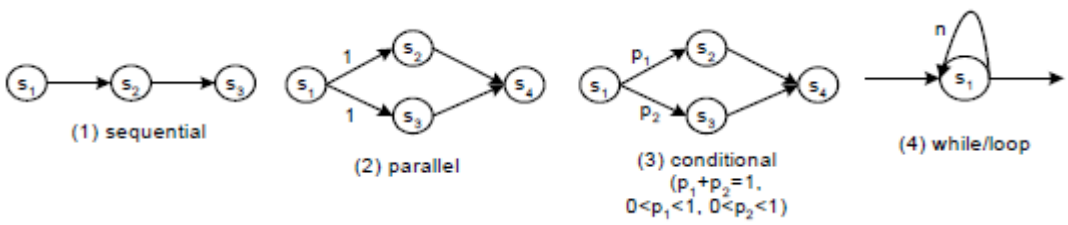

Fig. 3: Composition Flow Models

The selection of web services has been done by using the service composition models shown in the figure 3. In sequential method, service s3 is brought through by accessing services s2 and s1.

In parallel method, the service s4 is brought through two routes, one way is s1 → s2 →s4 and the other is s1 → s3 →s4.

In conditional method of service selection, the forthcoming service is selected based on the condition specified is correct or not. Where p1 and p2 are the conditions to be checked while accessing the services s2 and s3 respectively.

In the workflow-based composition methods, we should distinguish the static and dynamic workflow generation. The static one means that the requester should build an abstract process model before the composition planning starts. The abstract process model includes a set of tasks and their data dependency. Each task contains a query clause that is used to search the real atomic Web service to fulfill the task. In that case, only the selection and binding of atomic Web service is done automatically by program. The most commonly used static method is to specify the process model in graph. On the other hand, the dynamic composition both creates process model and selects atomic services automatically. This requires the requester to specify several constraints, including the dependency of atomic, the user's preference and so on.

EXPERIMENTAL EVALUATION

The effectiveness of the existing approach is shown by conducting three set of experiments:

- We evaluate the effectiveness of our results by utilizing f-measure in Semantic categorization of the web Services.
- Semantic similarity-based matching; we compute scores to rate the matching that are the average of a 10-pt precision recall curve.
- The overall time taken, measured in seconds, for service discovery. To be able to evaluate, we developed a prototype of our approach as to combine or integrate the services of different categories by using service composition method.

Additionally we also consider addition of related concepts to the data using ontology, deletion of irrelevant terms with and without adding new concepts. In particular, we consider the following data setup for clustering,

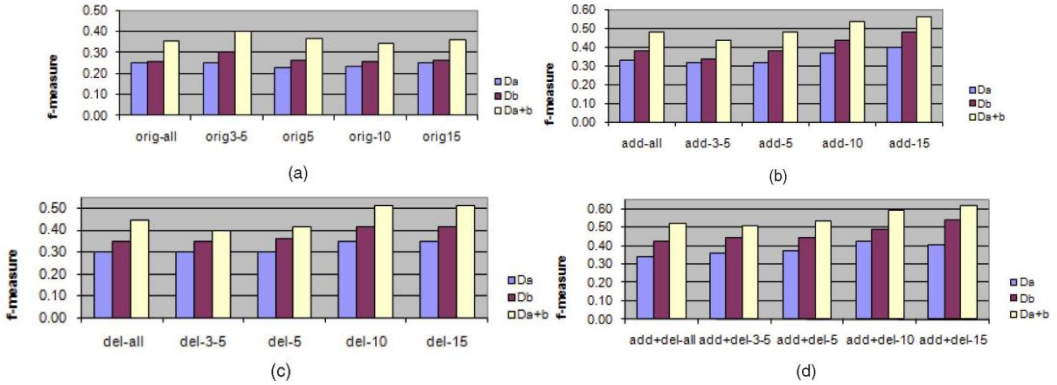

Fig. 10. (a) Experiment 1 — Orig. Setup. (b) Experiment 2 — Add. Setup (addition of ontology concepts to relevant terms of the service description vector). (c) Experiment 3 —Delete Setup. (d) Experiment 4 — Add and Delete Setup (addition of ontology concepts to all terms of the service description vector)

(1) **Orig.** – It is a baseline for further comparisons which includes all initial preprocessing techniques, i.e., stop list, stemming, and pruning.
(2) **Add** – This setup includes related concepts from the core ontology.
(3) **Delete** – This involves the removal of the irrelevant terms from the service vectors.
(4) **Add and Delete** – This technique is a combination of add and delete.
(5) **Integration/Composition** – This technique is to provide the enhanced output related to the request with additional services.

RELATED WORK

The classification process has similarities to our approach in terms of construction of term vectors with relevant words and utilizing a hierarchical clustering approach for achieving the best results. Our approach builds on this by;

(1) Including relevant semantic concepts based on semantic relationship ranking for expanding the domain coverage
(2) Deletion of non-relevant terms resulting in the reduction of noise and increase in the purity of the clusters.
(3) The refined method of finding semantic based web services has been integrated using one of the flow models shown in the fig. 3.

The usage of synonyms does not capture the overall semantics of the domain and application functionality. However, our approach utilizes concepts extracted from domain ontology. These extracted concepts account for relationships between the domain objects and provide a comprehensive coverage for the underlying semantics for both the domain and the application functionality.

CONCLUSION AND WORK

This paper has aimed to give an overview of recent progress in automatic Web services composition. We propose a four step model for Web services composition process. The composition model consists of service presentation, translation, process generation, evaluation and execution. Each step requires different languages, platforms and methods.

REFERENCES

[1] S. Abramsky. Proofs as processes. Theoretical Computer Science, 135(1):5–9, 1994.

[2] T. Andrews et al. Business Process Execution Language for Web Services (BPEL4WS) 1.1. Online: http://www-106.ibm.com/developerworks/webservices/library/ws-bpel, May 2003.

[3] G. Bellin and P. J. Scott. On the pi-calculus and Linear Logic. Theoretical Computer Science, 135(1):11–65, 1994.

[4] T. Bellwood et al. Universal Description, Discovery and Integration specification (UDDI) 3.0. Online: http://uddi.org/pubs/uddi-v3.00-published-20020719.htm.

[5] B. Benatallah, M. Dumas, M.-C. Fauvet, and F. Rabhi. Patterns and skeletons for parallel and distributed computing, chapter Towards Patterns of Web Services Composition, pages 265–296. Springer-Verlag, 2003.

[6] D. Box et al. Simple Object Access Protocol (SOAP) 1.1. Online: http://www.w3.org/TR/SOAP/, 2001.

[7] F. Casati, S. Ilnicki, and L. Jin. Adaptive and dynamic service composition in EFlow. In Proceedings of 12th International Conference on Advanced Information Systems Engineering (CAiSE), Stockholm, Sweden, June 2000. Springer Verlag.

[8] F. Casati, M. Sayal, and M.-C. Shan. Developing e-services for composing eservices. In Proceedings of 13th International Conference on Advanced Information Systems Engineering (CAiSE), Interlaken, Switzerland, June 2001. Springer Verlag

[9] R. Chinnici et al. Web Services Description Language (WSDL) 1.2. Online: http://www.w3.org/TR/wsdl/